

Generalizing Demonstrated Motions and Adaptive Motion Generation using an Invariant Rigid Body Trajectory Representation

Maxim Vochten¹, Tinne De Laet², and Joris De Schutter¹

Abstract—In programming by demonstration, generalization is necessary to apply demonstrated motions in novel situations. Many existing trajectory representations have poor generalization capabilities since they are built on trajectory coordinates that depend on the context in which the motion is recorded. In order to generalize, the user is typically required to perform a large number of varied demonstrations. This paper instead emphasizes the usefulness of an invariant trajectory representation to separate essential motion information from context-specific information of the recorded demonstrations. The invariants are interpreted as the control inputs of a dynamical system describing the evolution of the trajectory. New trajectories are generated for novel situations as the solution of a constrained optimal control problem in which context-specific information of the novel situation is encoded in the constraints. Results indicate how, starting from only a single demonstration, new trajectories can be generated in novel situations while maintaining similarity with the original demonstration. Invariance in trajectory representations therefore proves useful to reduce the number of necessary demonstrations to learn and apply new motions.

I. INTRODUCTION

Both motion recognition and motion generation applications require trajectory representations that are able to generalize from a limited set of demonstrated motions. In *motion recognition*, motion models constructed from demonstrated motions need to be as generic as possible in order to recognize motions in varying situations such as different viewpoints of the camera. Examples of motion recognition applications include motion classification [1], human intent estimation [2], and motion segmentation [3]. In *motion generation*, generalization is necessary to adapt the demonstrated motion to novel situations, such as a new starting location, a new target location, a different execution speed, or a different robot platform. The generated motion trajectories should also be as similar as possible to the human-demonstrated motions in order to keep the motions natural and interpretable by the human. Examples of motion generation using demonstrations include human-robot interaction and collaboration [4] and teaching object manipulation tasks [5].

This paper focuses on representing and generating *rigid body motion trajectories*. Rigid body motions include many interesting motions such as the motion of a manipulated rigid object, the end effector of a robot manipulator, and a human body segment. A rigid body motion is typically characterized by the rotation of the body and the translation of a chosen

reference point attached to the body. Many motion trajectory representations already exist to encode these trajectories [6], [7], [8], though they are often based only on the position and/or velocity coordinates of the chosen reference point. These coordinates depend on the context or situation in which the demonstrated motion is recorded, such as the chosen reference frame, the object's starting position, or the execution speed. This dependency on contextual information limits the capability of the trajectory representation to generalize. Generating similar trajectories is then typically only well defined close to the demonstrated motion or between different demonstrations by interpolation. If a motion needs to be generated in situations or contexts not covered by the demonstrations, the user is required to demonstrate in this new situation as well, which is not desired.

This paper's approach in contrast uses an invariant trajectory representation [9] to extract essential kinematic motion features. Context-specific information not inherent to the motion itself is removed from the trajectory. Previous work applied invariant rigid body trajectory representations for motion recognition [9], while this paper investigates their usefulness for generalization and adaptive motion generation. The approach is schematically represented in Figure 1. The demonstrated motion trajectory is first transformed to the corresponding invariant representation (Section III). The invariants are interpreted as the control inputs of a dynamical system governing the evolution of the moving object's trajectory. Adapted trajectories of the same motion are generated in novel situations by solving an optimal control problem with contextual information of the new situation encoded in the constraints of the optimization problem (Section IV). Deviation from the invariant motion features of the original demonstrations is minimized in the objective function in order to maintain similarity with the original demonstration. Experiments in Section V show the range of novel situations in which the motion trajectory can be successfully generated while maintaining similarity with what was demonstrated.

II. RELATED WORK

This section briefly reviews invariant trajectory representations mainly used in recognition up till now and provides an overview of current techniques for encoding trajectories in programming by demonstration.

A. Invariant trajectory representations

An invariant trajectory representation aims to extract intrinsic features from a given motion trajectory that remain

¹M. Vochten and J. De Schutter, Department of Mechanical Engineering, KU Leuven, 3001 Leuven, Belgium. maxim.vochten@kuleuven.be and joris.deschutter@kuleuven.be

²T. De Laet, Faculty of Engineering Science, KU Leuven, 3001 Leuven, Belgium. tinne.delat@kuleuven.be

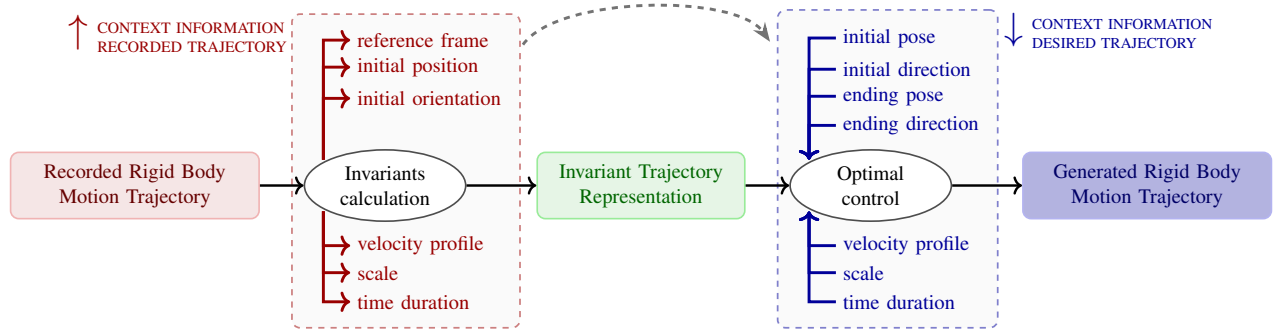


Fig. 1: Overview of the motion generation approach. Demonstrated motion trajectories are transformed to an invariant representation, separating and removing contextual information from the essential motion information contained in the invariants. Trajectories are generated by solving an optimal control problem in which the invariants serve as the control inputs of a dynamical system governing the evolution of the object’s trajectory. New contextual information is added in the form of constraints in the optimization problem in order to adapt the motion to novel situations.

unchanged under a set of transformations of the original trajectory. Such invariant representations have proven especially useful in motion recognition, since the extracted features can be made invariant with respect to the conditions during recording such as the viewpoint of the camera [1], [10]. This reduces the search space considerably during classification.

For representing 3D point trajectories, the differential geometric invariants of curvature and torsion (the *Frenet-Serret invariants* [11]) are often used. In [12], the curvature and torsion plus their first order derivatives are proposed for representing and recognizing point trajectories. These functions are view-invariant, since they only depend on the shape of the curve. Invariance with respect to time and execution speed is obtained by expressing these invariants as a function of a progress parameter along the trajectory instead of time.

Many invariant trajectory representations only describe the 3D translation of points. This is not sufficient for rigid body motion trajectories in which the orientation of the object is also important. The *extended Frenet-Serret invariants* for rigid body motion proposed in [9], [13] are a direct extension of the Frenet-Serret invariants for point trajectories, maintaining all the invariant properties of the Frenet-Serret invariants. A disadvantage however of the Frenet-Serret invariants is the dependency on the choice of reference point for the translation. To describe the complete 3D translation and rotation of a rigid body regardless of the choice of reference point, [14] defines differential invariants based on the motion of the instantaneous screw axis moving along with the object. Though the *screw axis-based invariants* of [14] possess more invariant properties than the extended Frenet-Serret invariants, the extended Frenet-Serret invariants may outperform them in practice, since including more and more invariant properties also increases the sensitivity to noise [9]. Therefore, this paper focuses on the extended Frenet-Serret invariants.

B. Trajectory encoding in programming by demonstration

One of the main challenges in programming by demonstration is generalizing from one or a few demonstrations to reproduce motions in different situations or contexts. Two main approaches exist to represent and generalize trajectories: probabilistic approaches and dynamical system approaches [15]. The separation between the two is not that strict though since for example the parameters in dynamical system approaches are often learned using probabilistic methods.

Probabilistic representations generalize demonstrated motions by modeling the variability in repeatedly executed demonstrations of the motion. *Hidden Markov Models* (HMM) are used in [16] to model the set of demonstrated trajectories, in which identified key points indicate the transition between the states in the HMM. Trajectories are reproduced by interpolating between these key points. In [8] a set of demonstrated trajectories is modeled using *Gaussian Mixture Modeling*. A smoothed generalized version of the trajectory is retrieved using *Gaussian Mixture Regression*. The *Probabilistic Movement Primitives* [7] represent trajectories as probability distributions allowing probabilistic operations on the underlying movement primitives such as blending or continuously switching from one motion primitive to the other. However, a problem with probabilistic representations is that multiple demonstrations are needed to calculate probability distributions and that extrapolating outside of the learned distribution of demonstrations yields poor results.

Dynamical system representations focus on providing robustness in the presence of perturbations such as changes in the target position or obstacles along the path. The *Dynamic Movement Primitives* [6] represent trajectories as dynamical systems consisting of the sum of a stable global attractor and a forcing term encoding the trajectory as a weighted sum of pre-defined basis functions. The global attractor provides the means to generalize to new target locations. These dynamical systems may however experience instability problems or convergence to other locations in state space when trying

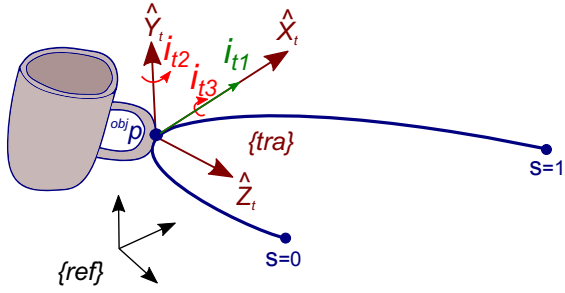


Fig. 2: The translation of a reference point ^{obj}p on the rigid body is fully described by the translational Frenet-Serret invariants i_{t1} , i_{t2} , and i_{t3} defined in a moving frame $\{tra\}$.

to reach a given target location. The *Stable Estimator of Dynamical Systems* in [17] on the other hand is designed to be globally stable. Even so, the generated trajectories lose more and more similarity with the demonstrated trajectories when moving away from the demonstrations in state space.

Some of the discussed trajectory representations have a level of built-in invariance such as time invariance by using a progress variable. To eliminate every contextual dependency however, a coordinate-free invariant trajectory representation is applied in the next section.

III. EXTENDED FRENET-SERRET INVARIANTS

To adequately generalize the motion trajectory of a rigid object, the contextual dependencies need to be eliminated such that only the essential motion information remains. To achieve this goal, the demonstrated trajectory is transformed to an invariant representation using the *extended Frenet-Serret invariants* [9]. The evolution of the object's trajectory is formulated here as a dynamical system in which the invariants function as the control inputs. Controlling this dynamical system is the basis on which the proposed trajectory generation procedure is built in Section IV.

A. Invariance w.r.t. choice reference frame and initial pose

To eliminate the dependency of the trajectory coordinates on the choice of a reference frame in which the coordinates are measured and the initial position and orientation (pose) from which the object starts moving, the *extended Frenet-Serret invariants* are calculated. These invariants consist of six functions: three for describing translation and three for rotation. The translation invariants are called the *translational Frenet-Serret invariants* and describe the spatial motion of a chosen reference point on the rigid object. In Figure 2, the reference point is chosen as a point on the handle of a mug. The translational Frenet-Serret invariants are then defined using the reference point's translational velocity vector \mathbf{v} and its first and second order time derivatives $\dot{\mathbf{v}}$ and $\ddot{\mathbf{v}}$ (the units are supplied in the brackets $[\cdot]$):

$$i_{t1}(t) = \|\mathbf{v}\| \left[\frac{m}{s} \right]; \quad i_{t2}(t) = \frac{\|\mathbf{v} \times \dot{\mathbf{v}}\|}{\|\mathbf{v}\|^2} \left[\frac{1}{s} \right], \quad (1)$$

$$i_{t3}(t) = \frac{((\mathbf{v} \times \dot{\mathbf{v}}) \cdot \ddot{\mathbf{v}}) \|\mathbf{v}\|}{\|\mathbf{v} \times \dot{\mathbf{v}}\|^2} \left[\frac{1}{s} \right]. \quad (2)$$

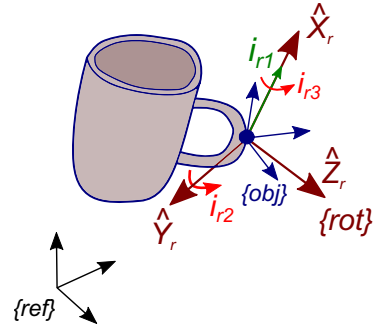


Fig. 3: The orientation of the object frame $\{obj\}$ is fully described by the rotational Frenet-Serret invariants i_{r1} , i_{r2} , and i_{r3} defined in a moving frame $\{rot\}$.

These invariants are most easily interpreted in a moving frame called the *translational Frenet-Serret frame* $\{tra\}$, attached to and moving along with the reference point. The first invariant i_{t1} is the magnitude of the translational velocity vector \mathbf{v} , which is oriented along the first axis $\hat{\mathbf{x}}_t$ of $\{tra\}$. The second invariant i_{t2} is the magnitude of the rotational velocity vector around the second axis $\hat{\mathbf{y}}_t$ in order to change the heading of the translational velocity vector \mathbf{v} . i_{t2} is closely related to the *curvature* κ of a space curve through the relation $i_{t2} = \kappa \|\mathbf{v}\|$. Intuitively, the curvature is the deviation of a curve with respect to a straight line. An increase in i_{t2} corresponds to a stronger turn in the trajectory. The third invariant i_{t3} is the magnitude of the rotational velocity vector around the first axis $\hat{\mathbf{x}}_t$ in order to change the orientation of the second axis $\hat{\mathbf{y}}_t$. The third invariant is closely related to the *torsion* τ of a space curve through the relation $i_{t3} = \tau \|\mathbf{v}\|$. Intuitively, the torsion allows the curve to move out of the plane spanned by $\hat{\mathbf{x}}_t$ and $\hat{\mathbf{z}}_t$. Note that while i_{t1} describes the velocity of the moving point along the trajectory, i_{t2} and i_{t3} describe the velocity of the moving frame itself. In other words, i_{t2} and i_{t3} represent the shape of the curve, while i_{t1} represents how fast we move along the curve.

The extended Frenet-Serret invariants extend the translational Frenet-Serret invariants with three rotational invariants describing the rotation of the rigid body. They are fully analogously defined as the translational invariants (1)-(2) but now using the rotational velocity vector $\boldsymbol{\omega}$:

$$i_{r1}(t) = \|\boldsymbol{\omega}\| \left[\frac{rad}{s} \right]; \quad i_{r2}(t) = \frac{\|\boldsymbol{\omega} \times \dot{\boldsymbol{\omega}}\|}{\|\boldsymbol{\omega}\|^2} \left[\frac{1}{s} \right], \quad (3)$$

$$i_{r3}(t) = \frac{((\boldsymbol{\omega} \times \dot{\boldsymbol{\omega}}) \cdot \ddot{\boldsymbol{\omega}}) \|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega} \times \dot{\boldsymbol{\omega}}\|^2} \left[\frac{1}{s} \right]. \quad (4)$$

These rotational invariants are also associated with the definition of a moving frame, called the *rotational Frenet-Serret frame* $\{rot\}$, illustrated in Figure 3.

The extended Frenet-Serret invariants are a coordinate-free description of the original trajectory, meaning that (1)-(4) do not depend on the choice of reference frame in which \mathbf{v} and $\boldsymbol{\omega}$ are originally expressed. The initial position and orientation also have no influence, since velocities and

derivatives of velocities are used. The resulting invariants are called *time-based*, since they are still time-dependent.

Reconstruction: Reconstructing the original trajectory corresponds to integrating a dynamical system $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$ in which the control inputs \mathbf{u} correspond to the invariants i and the states \mathbf{x} correspond to the object's position $^{obj}\mathbf{p}$ and the orientation of the moving frames $\{tra\}$, $\{rot\}$, and the object frame $\{obj\}$. To denote the orientations, the definition of *rotation matrix* is used. For example for the translational Frenet-Serret frame $\{tra\}$, the orientation with respect to the reference frame $\{ref\}$ is defined as: $^{tra}\mathbf{R} = [\hat{\mathbf{x}}_t \ \hat{\mathbf{y}}_t \ \hat{\mathbf{z}}_t]$.

The dynamic equations for translation consist of two parts. The first part contains the time evolution of the orientation of the translational Frenet-Serret frame $^{tra}\dot{\mathbf{R}}$, given by the Frenet-Serret differential equations [11]:

$$^{tra}\dot{\mathbf{R}}(t) = ^{tra}\mathbf{R}(t) \begin{bmatrix} 0 & 0 & i_{t2}(t) \\ 0 & 0 & -i_{t3}(t) \\ -i_{t2}(t) & i_{t3}(t) & 0 \end{bmatrix}. \quad (5)$$

Equation (5) specifies the change in the orientation $^{tra}\mathbf{R}$ of $\{tra\}$ caused by the rotational velocity vector $\mathbf{i}_t = [i_{t3} \ i_{t2} \ 0]^T$. This equation illustrates how only i_{t2} and i_{t3} are responsible for the motion of the moving frame, and thus fully determine the shape of the trajectory. The last matrix in (5) is a 3×3 skew-symmetric matrix constructed from \mathbf{i}_t . Using the skew-symmetric operator $[\cdot]_{\times}$, (5) is compactly notated as:

$$^{tra}\dot{\mathbf{R}}(t) = ^{tra}\mathbf{R}(t) [\mathbf{i}_t]_{\times}. \quad (6)$$

The second part of the dynamic equations for translation uses the result of the first part, the orientation of the Frenet-Serret frame $^{tra}\mathbf{R}$, in order to determine the time evolution of the object's position $^{obj}\dot{\mathbf{p}}$ as:

$$^{obj}\dot{\mathbf{p}}(t) = ^{tra}\mathbf{R}(t) \begin{pmatrix} i_{t1} \\ 0 \\ 0 \end{pmatrix}. \quad (7)$$

Equation (7) can also be written as:

$$^{obj}\dot{\mathbf{p}}(t) = i_{t1}\hat{\mathbf{x}}_t, \quad (8)$$

with $\hat{\mathbf{x}}_t$ the first axis of $\{tra\}$. The dynamic equations for translation are interpreted as follows: at time t , (8) tells us how fast we are traveling in the current traveling direction, while (6) tells us the change in traveling direction.

The dynamic equations for rotation are again very analogous to the case for translation. The time evolution of the rotational Frenet-Serret frame $\{rot\}$ is found similarly as in (5) with $\mathbf{i}_r = [i_{r3} \ i_{r2} \ 0]^T$:

$$^{rot}\dot{\mathbf{R}}(t) = ^{rot}\mathbf{R}(t) [\mathbf{i}_r]_{\times}. \quad (9)$$

The evolution of the object frame $\{obj\}$ is given by:

$$^{obj}\dot{\mathbf{R}}(t) = \left[^{rot}\mathbf{R}(t) \begin{pmatrix} i_{r1} \\ 0 \\ 0 \end{pmatrix} \right]_{\times} ^{obj}\mathbf{R}, \quad (10)$$

which is the rotational equivalent of (7). The rotational velocity vector ω contained in i_{r1} is first expressed in $\{ref\}$,

then transformed to a skew-symmetric matrix and finally multiplied with the original orientation $^{obj}\mathbf{R}$. (The skew-symmetric matrix is left-multiplied since the coordinates are expressed in $\{ref\}$). Equation (10) is compactly notated as:

$$^{obj}\dot{\mathbf{R}}(t) = [i_{r1}\hat{\mathbf{x}}_r]_{\times} ^{obj}\mathbf{R}, \quad (11)$$

with $\hat{\mathbf{x}}_r$ the first axis of $^{rot}\mathbf{R}$.

Given the orientations $^{obj}\mathbf{R}$, $^{rot}\mathbf{R}$, $^{tra}\mathbf{R}$, and the position $^{obj}\mathbf{p}$ at the start t_0 of the trajectory, the original trajectory is reconstructed by integrating (6), (8), (9), and (11).

B. Invariance w.r.t. motion execution style

The current *time-based invariants* still depend on the motion execution style. Motion execution style refers to how the user chooses to execute the motion: choice of the velocity profile along the trajectory, the duration of the motion, and the scale of the motion.

To eliminate these dependencies, the invariants are expressed as a function of a scalar progress parameter along the trajectory called the *degree of advancement*. Since translation and rotation are independent in the extended Frenet-Serret invariants, two degrees of advancement are defined here. The degree of advancement for translation is the normalized arc length $s(t)$ along the trajectory. The degree of advancement for rotation is the normalized rotated angle $\theta(t)$ along the trajectory. At the start of the trajectory t_0 , s and θ are equal to zero and at the end of the trajectory t_f , s and θ are equal to one. $s(t)$ and $\theta(t)$ will be defined by their derivatives $\dot{s}(t)$ and $\dot{\theta}(t)$, also called the *rates of advancement*:

$$\dot{s}(t) = \frac{\|\mathbf{v}\|}{L} \begin{bmatrix} 1 \\ s \end{bmatrix}; \quad \dot{\theta}(t) = \frac{\|\omega\|}{\Theta} \begin{bmatrix} 1 \\ \theta \end{bmatrix}, \quad (12)$$

in which the scaling factors L and Θ are chosen as:

$$L = \int_{t_0}^{t_f} \|\mathbf{v}\| dt; \quad \Theta = \int_{t_0}^{t_f} \|\omega\| dt, \quad (13)$$

so that L is the total traversed distance and Θ is the total rotated angle. Each time-based descriptor component $i_k(t)$ is re-parameterized to its geometric counterpart by substituting t with s or θ and is subsequently divided by the corresponding rate of advancement, as in [14]:

$$I_{tk}(s) = \frac{i_{tk}(t(s))}{\dot{s}(t(s))}; \quad I_{rk}(\theta) = \frac{i_{rk}(t(\theta))}{\dot{\theta}(t(\theta))}, \quad (14)$$

($k = 1, 2, 3$). I_{t1} and I_{r1} turn out to be constant and equal to L and Θ respectively, signifying a constant progression along the trajectory (independent of the velocity profile). The other invariants I_{t2} , I_{t3} , I_{r2} , and I_{r3} have all been made dimensionless by the division with the rate of advancement, and are thus called *dimensionless geometric*.

Reconstruction: The dynamical system equations for the *time-based invariants* (6), (8), (9), and (11) can similarly be formulated for the *dimensionless geometric invariants*. The evolution of the moving frames $\{tra\}$ and $\{rot\}$ as a function of the degrees of advancement s and θ is then:

$$^{tra}\mathbf{R}'(s) = ^{tra}\mathbf{R}(s) [\mathbf{I}_t]_{\times}, \quad (15)$$

$$^{rot}\mathbf{R}'(\theta) = ^{rot}\mathbf{R}(\theta) [\mathbf{I}_r]_{\times}, \quad (16)$$

with $\mathbf{I}_t = [I_{t3} \ I_{t2} \ 0]^T$ and $\mathbf{I}_r = [I_{r3} \ I_{r2} \ 0]^T$. (The derivatives with respect to s and θ are notated with $[\cdot]'$.) The evolution of the object's position and orientation is given by:

$$^{obj}\mathbf{p}'(s) = L\hat{\mathbf{x}}_t, \quad (17)$$

$$^{obj}\mathbf{R}'(\theta) = [\Theta\hat{\mathbf{x}}_r]_{\times} ^{obj}\mathbf{R}(\theta). \quad (18)$$

After reconstruction, $^{obj}\mathbf{p}(s)$ and $^{obj}\mathbf{R}(\theta)$ are reparameterized to $^{obj}\mathbf{p}(t)$ and $^{obj}\mathbf{R}(t)$ using the degrees of advancement $s(t)$ and $\theta(t)$ to again obtain the original rigid body trajectory as a function of time.

IV. TRAJECTORY GENERATION

The dynamical system equations (15)-(18) allow reconstructing the demonstrated trajectory from the dimensionless geometric invariants. In order to generate the motion in novel situations, the dynamical system equations are embedded in a constrained optimal control problem with the contextual information of the novel situation encoded in the constraints.

The states \mathbf{x} of the optimal control problem are the object position $^{obj}\mathbf{p}(s)$, the object orientation $^{obj}\mathbf{R}(\theta)$, the translational Frenet-Serret frame orientation $^{tra}\mathbf{R}(s)$, and the rotational Frenet-Serret frame orientation $^{rot}\mathbf{R}(\theta)$. Note that the states are parameterized by the degrees of advancement s and θ . This decouples the determination of the geometric shape of the trajectory by the optimal control problem from the determination of the velocity profile along the trajectory, reducing the number of degrees of freedom in the optimization problem. After solving the optimal control problem, the desired motion profiles $s(t)$ and $\theta(t)$ need to be supplied to again express $^{obj}\mathbf{p}(s)$ and $^{obj}\mathbf{R}(\theta)$ as a function of time.

The control inputs to the dynamical system are the translational invariants $\mathbf{I}_t = [I_{t3} \ I_{t2} \ 0]^T$ and the rotational invariants $\mathbf{I}_r = [I_{r3} \ I_{r2} \ 0]^T$. Two parameters are also sought: the total traversed distance L and the total rotated angle Θ .

The optimal control problem is formulated as follows:

$$\underset{\mathbf{x}, \mathbf{I}_t, \mathbf{I}_r, L, \Theta}{\text{minimize}} \int_0^1 \|\mathbf{I}_t(s) - \mathbf{I}_t^d(s)\|_{W_t}^2 ds + \int_0^1 \|\mathbf{I}_r(\theta) - \mathbf{I}_r^d(\theta)\|_{W_r}^2 d\theta \quad (19)$$

subject to:

$$^{tra}\mathbf{R}'(s) = ^{tra}\mathbf{R}(s) [\mathbf{I}_t]_{\times} \quad (20)$$

$$^{rot}\mathbf{R}'(\theta) = ^{rot}\mathbf{R}(\theta) [\mathbf{I}_r]_{\times} \quad (21)$$

$$^{obj}\mathbf{p}'(s) = L\hat{\mathbf{x}}_t \quad (22)$$

$$^{obj}\mathbf{R}'(\theta) = [\Theta\hat{\mathbf{x}}_r]_{\times} ^{obj}\mathbf{R} \quad (23)$$

$$^{tra}\mathbf{R}(0) = ^{tra}\mathbf{R}_0, \ ^{obj}\mathbf{p}(0) = ^{obj}\mathbf{p}_0 \quad (24)$$

$$^{tra}\mathbf{R}(1) = ^{tra}\mathbf{R}_1, \ ^{obj}\mathbf{p}(1) = ^{obj}\mathbf{p}_1 \quad (25)$$

$$^{rot}\mathbf{R}(0) = ^{rot}\mathbf{R}_0, \ ^{obj}\mathbf{R}(0) = ^{obj}\mathbf{R}_0 \quad (26)$$

$$^{rot}\mathbf{R}(1) = ^{rot}\mathbf{R}_1, \ ^{obj}\mathbf{R}(1) = ^{obj}\mathbf{R}_1 \quad (27)$$

Objective function (19) contains the deviation of the invariants of the generated trajectory from the invariants of the demonstrated trajectory \mathbf{I}^d . By minimizing this deviation, the invariant features of the original demonstrated trajectory

are preserved, ensuring similarity in shape between the generated and demonstrated trajectories. The supplied weighting matrices W_t and W_r weight the influence from one invariant with respect to the other. The weighting matrices could alternatively also hold statistical information if multiple demonstrations are available. By setting the weighting matrices equal to the inverse of the covariance matrix along the trajectory, invariant features with low variance get a higher weight and are more likely to be maintained in the reconstruction.

Dynamic constraints (20)-(23) express the evolution of the object's position and orientation as dictated by the invariant control inputs \mathbf{I}_t and \mathbf{I}_r , and the parameters L and Θ .

Initial constraints (24) and (26) specify the initial position and orientation of the object along with the initial orientation of the Frenet-Serret frames. The latter orientations specify among other the initial direction in which the object starts moving. The initial position and orientation of the object are typically known, while the initial direction is known if the object is already in motion, otherwise it may be omitted.

Ending constraints (25) and (27) specify the final position and orientation of the object along with the ending orientation of the Frenet-Serret frames. The latter orientations specify among other the final direction in which the object arrives at the target location. If the final direction is not important this constraint may be omitted. While in (25) and (27) these ending constraints are expressed as hard constraints, it is also possible to formulate them as soft constraints in which a weight can for example indicate uncertainty on the target's exact location.

The trajectory generation approach covers both point-to-point motions and periodic motions. For the latter, the starting and end constraints should be set equal to each other.

Extra constraints such as obstacle avoidance can always be added. However this is not in the scope of this paper.

V. EXPERIMENTS

The generalization capabilities of the proposed trajectory generation approach are evaluated by showing how measured rigid body motions are adapted to novel situations after solving the optimal control problem of the previous section. Furthermore, the shape-similarity of generated trajectories is qualitatively compared with another trajectory representation, the dynamic movement primitives [6].

A. From measurements to invariant representation

The rigid body motion of a human-manipulated rigid object (coffee mug) is recorded by tracking markers attached to the object using a marker-based vision system (Krypton K600 from NIKON Metrology). Figure 4 shows examples of the recorded rigid body motions. The reference frame corresponds to the reference frame of the camera. The reference point for the translation of the rigid object is chosen as the average position of all the markers.

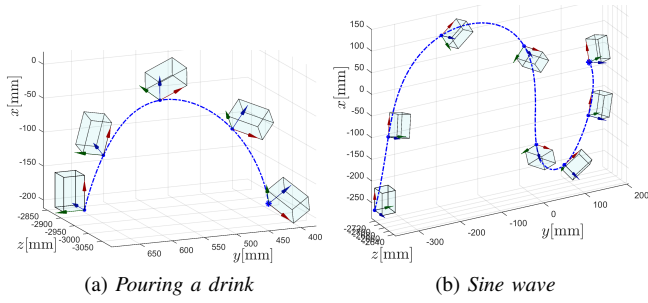


Fig. 4: Visualization of some of the recorded rigid body motions after pre-processing the marker coordinates. The blue line represents the translation of the chosen reference point, while the arrows and cuboids indicate the object orientation.

In a pre-processing step, explained in more detail in [9], the measured marker trajectories are first smoothed to reduce noise. Afterwards, the translational and rotational velocity of the rigid body are calculated in a numerically stable way along with their first and second order derivatives. Given the velocities and the velocities' derivatives, the equations in Section III are used to calculate the invariant representation.

B. Solving the optimal control problem

Generating new trajectories requires solving the optimal control problem defined in Section IV. The free and open-source software CasADi [18] is used for this. CasADi provides general purpose functionalities and building blocks for nonlinear numerical optimization and optimal control. A direct multiple shooting method [19] is applied to transform the optimal control to an equivalent nonlinear optimization problem by discretization of both the states \mathbf{x} and the controls $(\mathbf{I}_t, \mathbf{I}_r)$. The dynamical system equations are integrated over the discretization intervals using a Runge-Kutta integrator. The resulting sparse nonlinear program (NLP) is fed to a low-level numerical solver (IPOPT) which uses Sequential Quadratic Programming to solve the NLP. The states and invariants are initialized with those of the demonstration.

C. Adaptive trajectory generation in different contexts

The generalization capabilities are illustrated for the *pouring* motion shown in Figure 4. Starting from this single demonstration, the motion trajectory is adapted to different situations. The optimal control approach in Section IV only generates the geometric shape of the trajectory, so the motion profiles $s(t)$ and $\theta(t)$ of the original demonstration were applied to the solution to express the object's position and orientation again as a function of time. Other motion profiles could just as well be chosen to adjust the timing along the trajectory.

1) *Reconstruction of original trajectory:* Figure 5 shows how the original trajectory can be reconstructed from the invariant representation. The initial and final position and

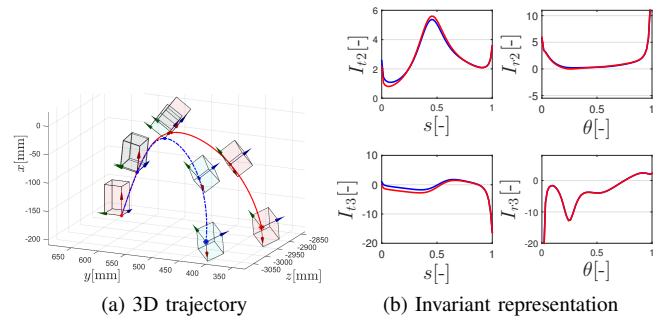


Fig. 6: Adaptation (red) of the original *pouring* trajectory (blue) to a new target position. The trajectory is adjusted by a small change in the translational invariants, maintaining similarity in shape.

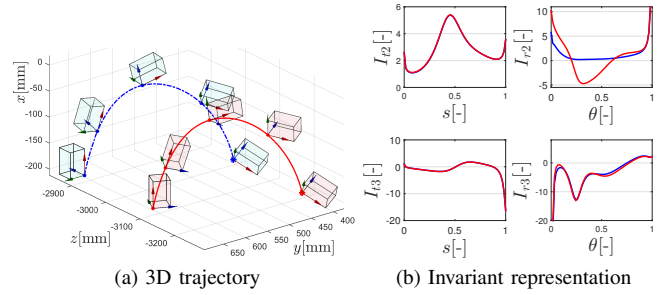


Fig. 7: Adaptation (red) of the original *pouring* trajectory (blue) to a new starting orientation of the object with the same target position and target orientation. A change in the rotational invariants adapts the object's orientation. (To more easily compare differences between orientations, a constant displacement along the z -axis was added between the two trajectories.)

orientation are constrained to be the same as the demonstration. There remains a small error in the trajectory coordinates between the start and the end due to small errors in the invariant calculation step and due to integration errors while solving the optimal control problem.

2) *Different target position:* To adapt the *pouring* motion to a different target position (e.g. another glass to pour into), the end constraint on the position is adjusted in (25). Figure 6 shows how the resulting trajectory is adapted by a slight change in the translational invariants. The invariants deviate mostly at the start since that is where changes in the invariants have the largest impact on the rest of the trajectory. Since the change in the invariants is small, shape-similarity with the original trajectory is maintained.

3) *Different starting orientation:* Figure 7 illustrates how the trajectory is adapted due to a change in the starting orientation of the object (e.g. after rotating the coffee mug). A change in the rotational invariants is now necessary to bring the object to the same end orientation.

4) *On-line adaptation:* While the whole trajectory is generated at once by solving the optimal control problem,

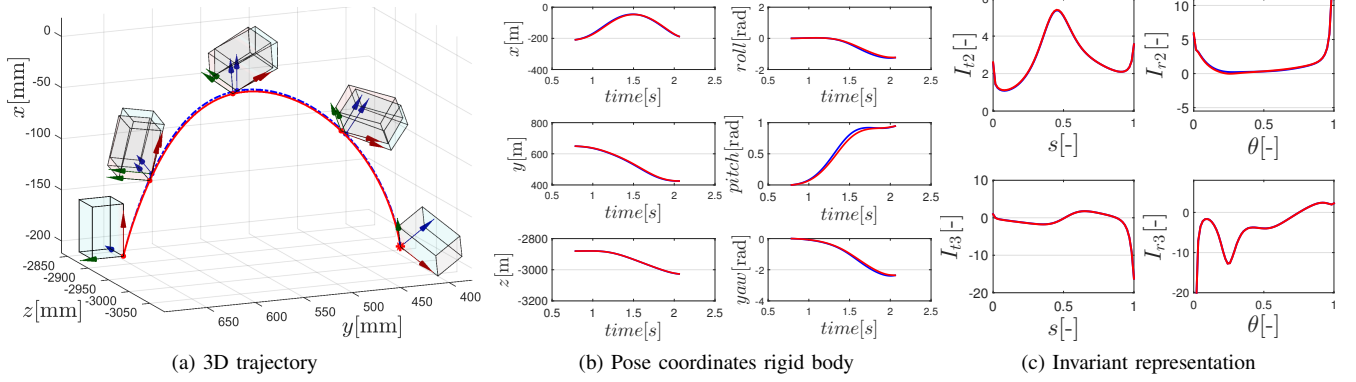


Fig. 5: Reconstruction (red) of the original demonstrated *pouring* trajectory (blue) starting from the invariant representation.

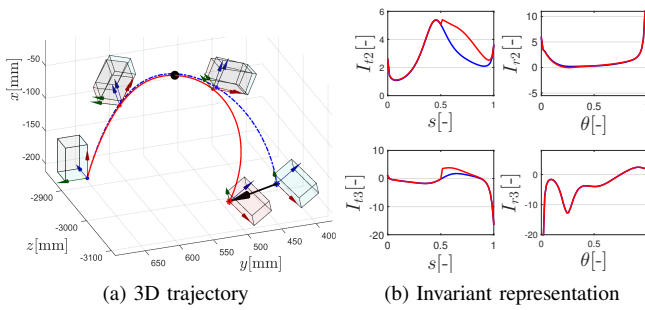


Fig. 8: On-line adaptation (red) of the original *pouring* trajectory (blue) to a new target position after 50% translation (black dot) along the original trajectory. The translational invariants deviate after $s = 0.5$ to address the change in target position.

it is also possible to generate updated trajectories during motion if there is a change or perturbation in the environment during motion. Figure 8 shows what happens if after 50% of the translation of the original trajectory (indicated by the black dot), the target position is suddenly changed to another position. The trajectory is now smoothly adapted to the new target location maintaining continuity in the trajectory, since the current position, orientation and traveling direction are all constrained in (24) and (26). This means there are no sudden jumps in the generated trajectory.

D. Comparison with dynamic movement primitives

The invariant representation is compared with another trajectory representation, the dynamic movement primitives [6]. The dynamic movement primitives represent trajectories as a dynamical system consisting of a global attractor to the target position and a non-linear force term encoding the trajectory shape. Figure 9 compares the shape between the generated trajectories of the invariant representation and those of the dynamic movement primitives for which the parameters, as defined in [6], are chosen as $\alpha_z = 3$, $\beta_z = \alpha_z/4$, $\alpha_x = \alpha_z/3$, and $N = 10$. New trajectories (red) are generated for four new target positions. While the global attractor

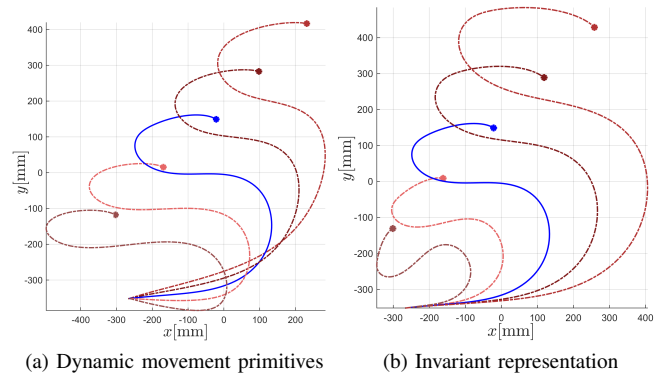


Fig. 9: The generated trajectories (red) of the invariant representation appear more similar in shape to the demonstration (blue), compared to the dynamic movement primitives.

of the dynamic movement primitives succeeds in helping to converge to the new target position, it also distorts the trajectory shape in the process. The invariant representation on the other hand much better maintains similarity in shape with respect to the original demonstration.

VI. DISCUSSION

Though the invariant representation is advantageous for generalizing motion trajectories in motion recognition and motion generation thanks to its invariant properties, there are also some important remarks to be made:

Robustness to noise: Equations (1)-(4) require the calculation of the translational and rotational velocity and their first and second order derivatives. Since higher order derivatives are more sensitive to measurement noise, care must be taken not to amplify the noise. This can be done by applying a smoother on the recorded data, e.g. a Kalman smoother.

Singularities: Singularities are points along the trajectory where one or more of the invariants in (1)-(4) along with their corresponding moving Frenet-Serret frame are not (uniquely) defined. For the translation, this occurs when the object is not moving ($\|\mathbf{v}\| = 0$) or when the curve is locally a straight

line ($\|\mathbf{v} \times \dot{\mathbf{v}}\| = 0$). The latter case could be solved in theory by using an alternative definition of moving frame instead of the Frenet-Serret frame such as the Bishop frame [20]. This is a subject for further research. However in practice this was not a problem since due to the discretization of the trajectory, $\|\mathbf{v} \times \dot{\mathbf{v}}\|$ never becomes exactly zero.

Trajectory calculation time: Solving optimization problems generally requires multiple iterations in order to converge to a solution. This requires more calculation time than a dynamical system approach. Even so, the generated trajectories are calculated fast enough to be considered for on-line use. The trajectory in Figure 6 for example consisting of 64 samples, requires 4 iterations with a calculation time of 55 ms per iteration on an Intel Core i7-3740QM processor with 16GB RAM. Different options are still available to improve the calculation speed such as C-code generation and exploiting the special structure of the objective function (non-linear least squares problem).

Accuracy dynamic system integration: In the experiments a standard Runge-Kutta integrator is used to integrate the rotation matrices in the dynamic system equations (20-23). This integrator however does not guarantee preservation of the special orthogonal structure of the rotation matrix. In the experiments this effect was negligible however because of the relatively short horizons over which the integration occurs. Possible solutions to explicitly retain orthogonality include using special group integrators [21], Baumgarte stabilization in the dynamic constraints [22] or projecting the solution to an orthogonal basis in a post-processing step.

VII. CONCLUSION

A trajectory generalization and adaptation approach based on an invariant trajectory representation was developed. Since the invariants are intrinsic features of the trajectory, it is possible to generalize demonstrations even when only a single demonstration is available. Solving an optimal control problem allows to adapt the learned trajectory to novel situations while minimizing changes to the invariant motion features. This ensures that similarity with the original demonstration is maintained. The proposed trajectory generation approach can be applied to any rigid body motion. Future work includes implementing obstacle avoidance and using the approach for learning specific tasks on a robot manipulator.

SOFTWARE

The code for generating trajectories using the invariant representation has been made publicly available [23].

ACKNOWLEDGMENT

All authors gratefully acknowledge the financial support by the Flemish FWO project G0B6613.

REFERENCES

- [1] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "View-invariant motion trajectory-based activity classification and recognition," *Multimedia Systems*, 2006.
- [2] D. Aarno and D. Kragic, "Motion intention recognition in robot assisted applications," *Robotics and Autonomous Systems*, vol. 56, no. 8, pp. 692–705, 2008.
- [3] D. Kulic, W. Takano, and Y. Nakamura, "Online segmentation and clustering from continuous observation of whole body motions," *Robotics, IEEE Transactions on*, vol. 25, no. 5, pp. 1158–1166, 2009.
- [4] G. Maeda, M. Ewerton, R. Lioutikov, H. Ben Amor, J. Peters, and G. Neumann, "Learning interaction for collaborative tasks with probabilistic movement primitives," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. IEEE, 2014, pp. 527–534.
- [5] J. Aleotti and S. Caselli, "Robust trajectory learning and approximation for robot programming by demonstration," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 409–413, 2006.
- [6] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.
- [7] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems*, 2013, pp. 2616–2624.
- [8] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 2, pp. 286–298, 2007.
- [9] M. Vochten, T. De Laet, and J. De Schutter, "Comparison of rigid body motion trajectory descriptors for motion representation and recognition," in *2015 IEEE International Conference on Robotics and Automation*, 2015.
- [10] C. Rao, A. Yilmaz, and M. Shah, "View-invariant representation and recognition of actions," *International Journal of Computer Vision*, vol. 50, no. 2, pp. 203–226, 2002.
- [11] W. Kühnel, *Differential geometry: curves-surfaces-manifolds*. American Mathematical Soc., 2006, vol. 16.
- [12] S. Wu and Y. F. Li, "Flexible signature descriptions for adaptive motion trajectory representation, perception and recognition," *Pattern Recognition*, vol. 42, no. 1, pp. 194–214, 2009.
- [13] R. Soloperto, M. Saveriano, and D. Lee, "A bidirectional invariant representation of motion for gesture recognition and reproduction," in *2015 IEEE International Conference on Robotics and Automation*, 2015.
- [14] J. De Schutter, "Invariant description of rigid body motion trajectories," *Journal of Mechanisms and Robotics*, vol. 2, no. 1, pp. 011004/1–9, 2010.
- [15] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer handbook of robotics*. Springer, 2008, pp. 1371–1394.
- [16] A. Vakanski, I. Mantegh, A. Irish, and F. Janabi-Sharifi, "Trajectory learning for robot programming by demonstration using hidden markov model and dynamic time warping," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 4, pp. 1039–1052, 2012.
- [17] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *Robotics, IEEE Transactions on*, vol. 27, no. 5, pp. 943–957, 2011.
- [18] J. Andersson, "A General-Purpose Software Framework for Dynamic Optimization," PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium, October 2013.
- [19] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast motions in biomechanics and robotics*. Springer, 2006, pp. 65–93.
- [20] R. L. Bishop, "There is more than one way to frame a curve," *American Mathematical Monthly*, pp. 246–251, 1975.
- [21] A. M. Bloch, I. I. Hussein, M. Leok, and A. K. Sanyal, "Geometric structure-preserving optimal control of a rigid body," *Journal of Dynamical and Control Systems*, vol. 15, no. 3, pp. 307–330, 2009.
- [22] S. Gros, M. Zanon, M. Vukob, and M. Diehl, "Nonlinear mpc and mhe for mechanical multi-body systems with application to fast tethered airplanes," in *Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference, Noordwijkerhout, The Netherlands*, 2012, p. 143.
- [23] M. Vochten, "Rigid Body Motion Trajectory Descriptors," 2015. [Online]. Available: <http://people.mech.kuleuven.be/~u0091864/>